Model Development and Analysis of Clean & Efficient Engine Combustion

Russell Whitesides (PI),
Nick Killingsworth, Guillaume Petitpas, & Matthew McNenly

Lawrence Livermore National Laboratory

2015 DOE Hydrogen and Fuel Cells Program and Vehicle Technologies Office Annual Merit Review and Peer Evaluation Meeting

June 9, 2014 - Washington, DC

This presentation does not contain any proprietary, confidential, or otherwise restricted information.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC
LLNL-PRES-669646

Overview

Timeline

- Ongoing project with yearly direction from DOE

Barriers

- Inadequate understanding of the fundamentals of HECC
- Inadequate understanding of the fundamentals of mixed mode operation
- Computational expense of HECC simulations

Budget

- FY14 funding: $475K
- FY15 funding: $508K

Partners

- AEC Working Group:
  - Sandia NL, Cummins
- University:
  - UC Berkeley
- Industrial:
  - Convergent Science Inc.
  - Nvidia

Relevance – Enhanced understanding of HECC requires expensive models that fully couple detailed kinetics with CFD

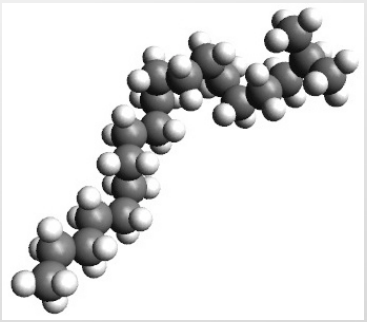
Objective

Create faster and more accurate combustion solvers.

- Accelerates R&D on three major challenges identified in the VT multi-year program plan:
  - A. *Lack of fundamental knowledge of advanced engine combustion regimes*
  - C. *Lack of modeling capability for combustion and emission control*
  - D. *Lack of effective engine controls*

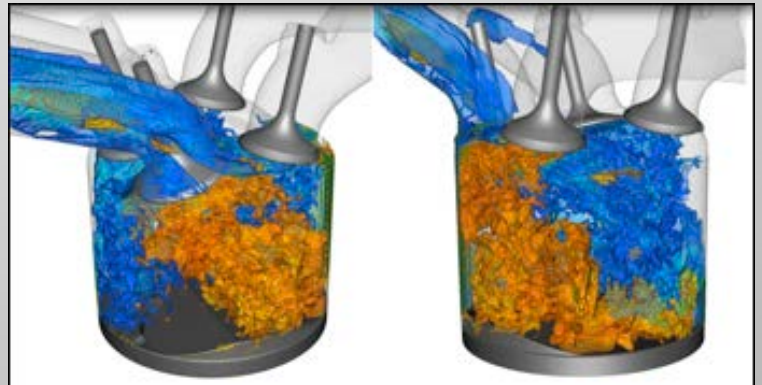
We want to use…

Detailed chemistry



Ex. Biodiesel component $C_{20}H_{42}$ (LLNL)
7.2K species
53K reaction steps

in highly resolved 3D simulations



Ex. SI/HCCI transition ~10M cells for Bosch in LLNL's hpc4energy incubator

Accurate simulations yield improved engine designs

Objective: Enhance understanding of clean and efficient engine operation through detailed numerical modeling

n-alkanes

n-hexadecane ($C_{16}H_{34}$)

n-octadecane ($C_{18}H_{38}$)

n-eicosane ($C_{20}H_{42}$)

naphtho-aromatic

tetralin ($C_{10}H_{12}$)

iso-alkane

2,2,4,4,6,8,8-heptamethylnonane ($C_{16}H_{34}$)

aromatics

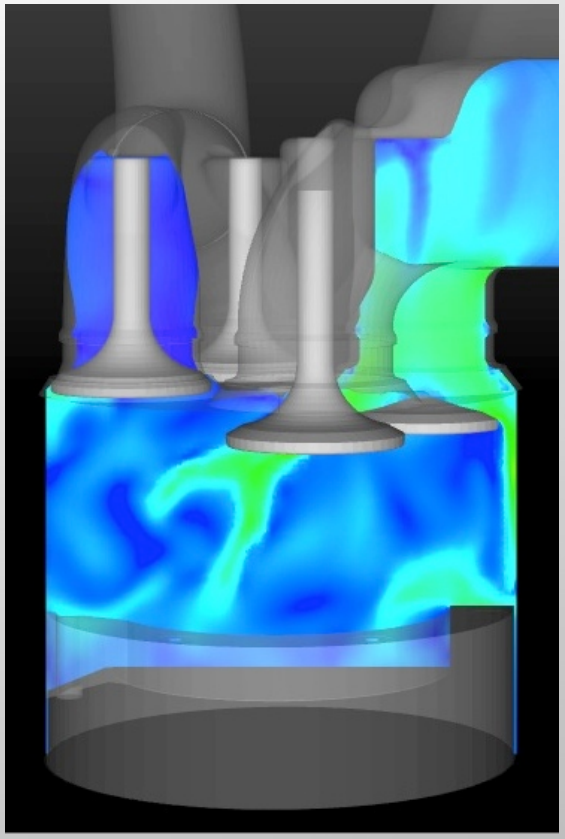
cyclo-alkanes

1,2,4-trimethylbenzene (C_9H_{12})

n-butylcyclohexane ($C_{10}H_{20}$)

1-methylnaphthalene ($C_{11}H_{10}$)

trans-decalin ($C_{10}H_{18}$)



Representative Detailed Chemical Kinetics

High-Fidelity Fluid Mechanics

Accurate simulations yield improved engine designs

Approach: Develop analysis tools leading to clean, efficient engines in collaboration with industry, academia and national labs

- Gain fundamental and practical insight into HECC regimes through numerical simulations and experiments
- Develop and apply numerical tools to simulate HECC by combining multidimensional fluid mechanics with chemical kinetics
- Reduce computational expense for HECC simulations
- Make accurate and efficient models accessible to industry
- Democratize simulation: bring chemical kinetics-fluid mechanics computational tools to the desktop PC

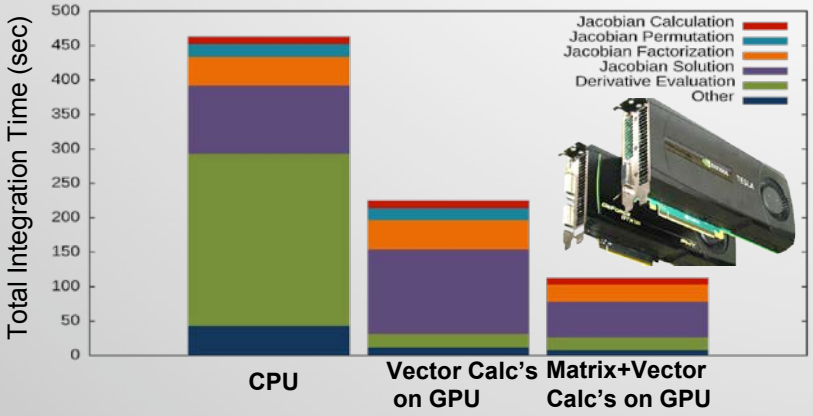
Enable more accurate simulations via more detailed physical representation

Milestones: We are developing and validating detailed engine and combustion modeling tools

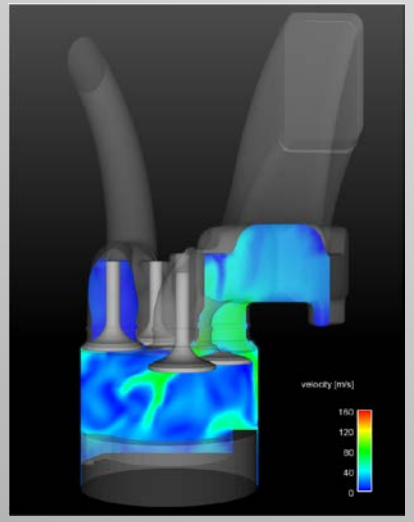
- ✓ Fast detailed chemistry for CFD:
  - ✓ Improved CPU/GPU solver for “Engineering” size mech.
  - ✓ Implemented CPU/GPU parallel chemistry work-sharing
  - ✓ Modularized multi-zone
- ✓ Uncertainty quantification in HCCI simulations
- Simulations of surrogate diesel engine experiments

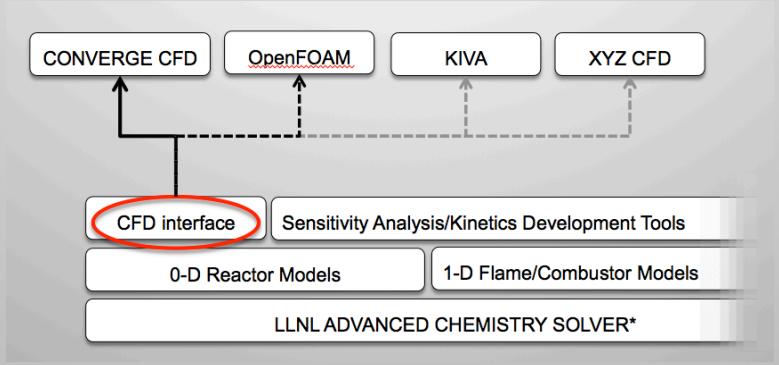
We are on track.

FY2014 Accomplishments



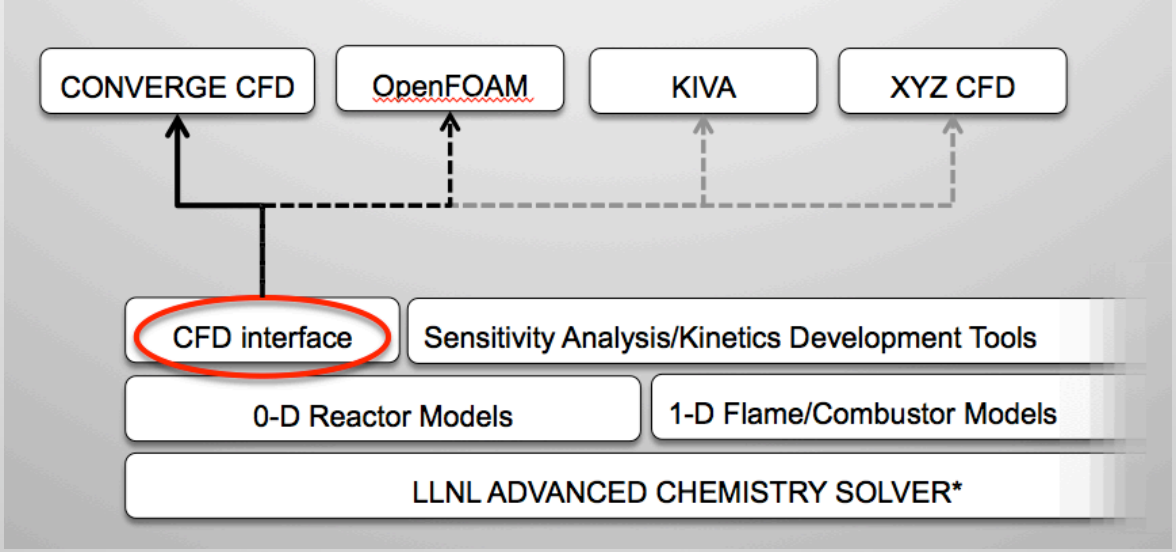
- GPU chemistry proven for *5x* speedup over CPU calculations for large mechanisms
- Highly resolved simulations of diesel engine including intake and exhaust manifolds
- HCCI/PCCI simulations with detailed chemistry with agreement to experiment
- Developed general chemistry interface for coupling to CFD packages with operator split chemistry method





Significant achievements in simulation performance and efficiency.

Technical Accomplishment: Improvements to Fast Chemistry Solver for CFD

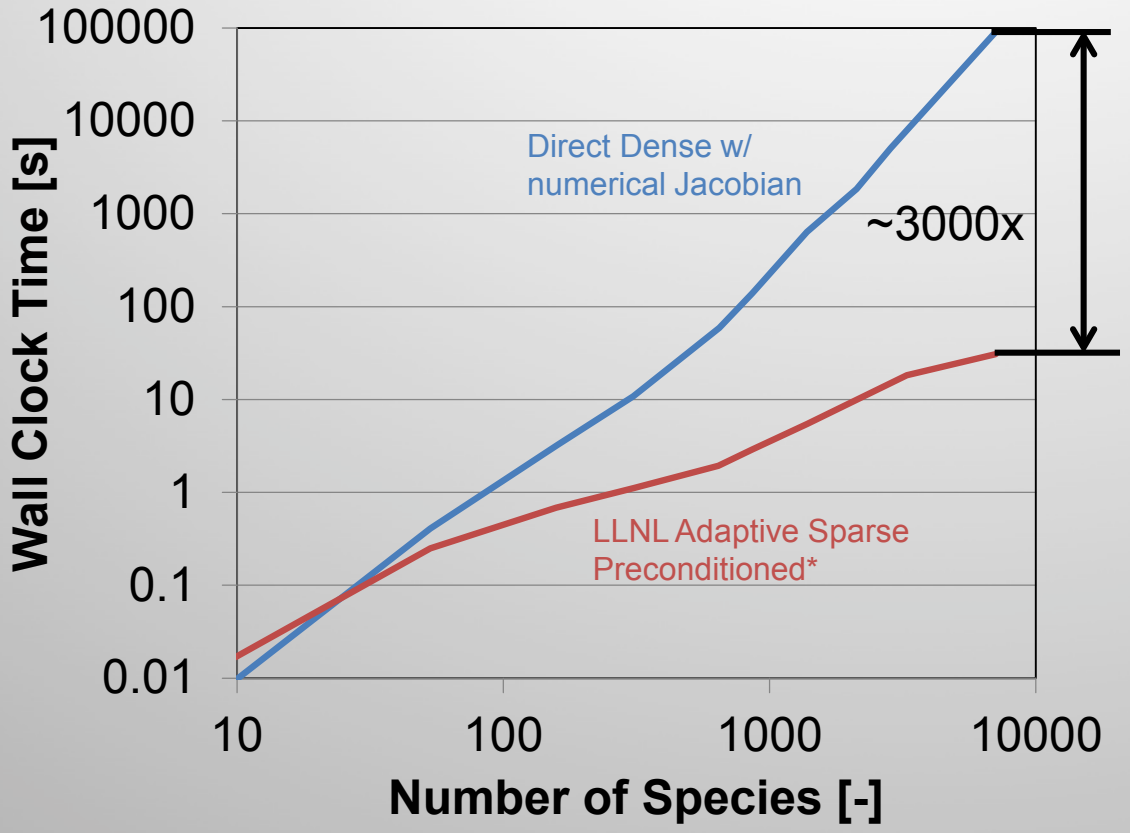


- Improvements for “engineering” size mechanisms (CPU/GPU)
- Improvements for large mechanisms on GPU (2-4x faster matrix math)
- Work-sharing for improved parallel scaling (CPU/GPU)
- Modular multi-zone capability

*ACE076: McNenly (PI)

Leveraging advanced solver work for practical engineering simulations

Technical Accomplishment: CPU and GPU Speedups for "Engineering" Size Mechanisms

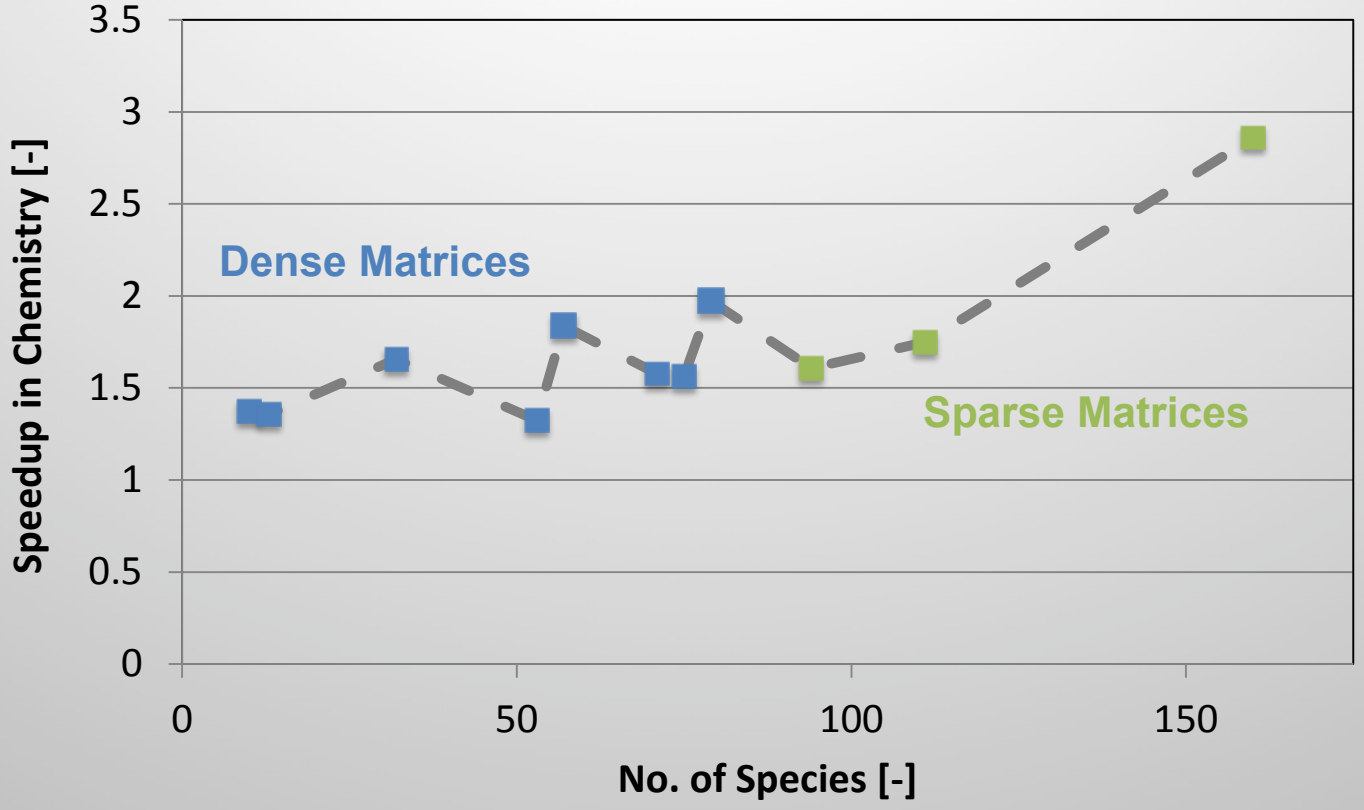


- Large mechanisms still unaffordable for many CFD scenarios.
- Can we reduce simulation times for smaller mechanisms?
- Apply techniques learned from adaptive sparse work to small mechanism approach.

*ACE076: McNenly (PI)

Previous work focused on large (> 100 species) mechanisms

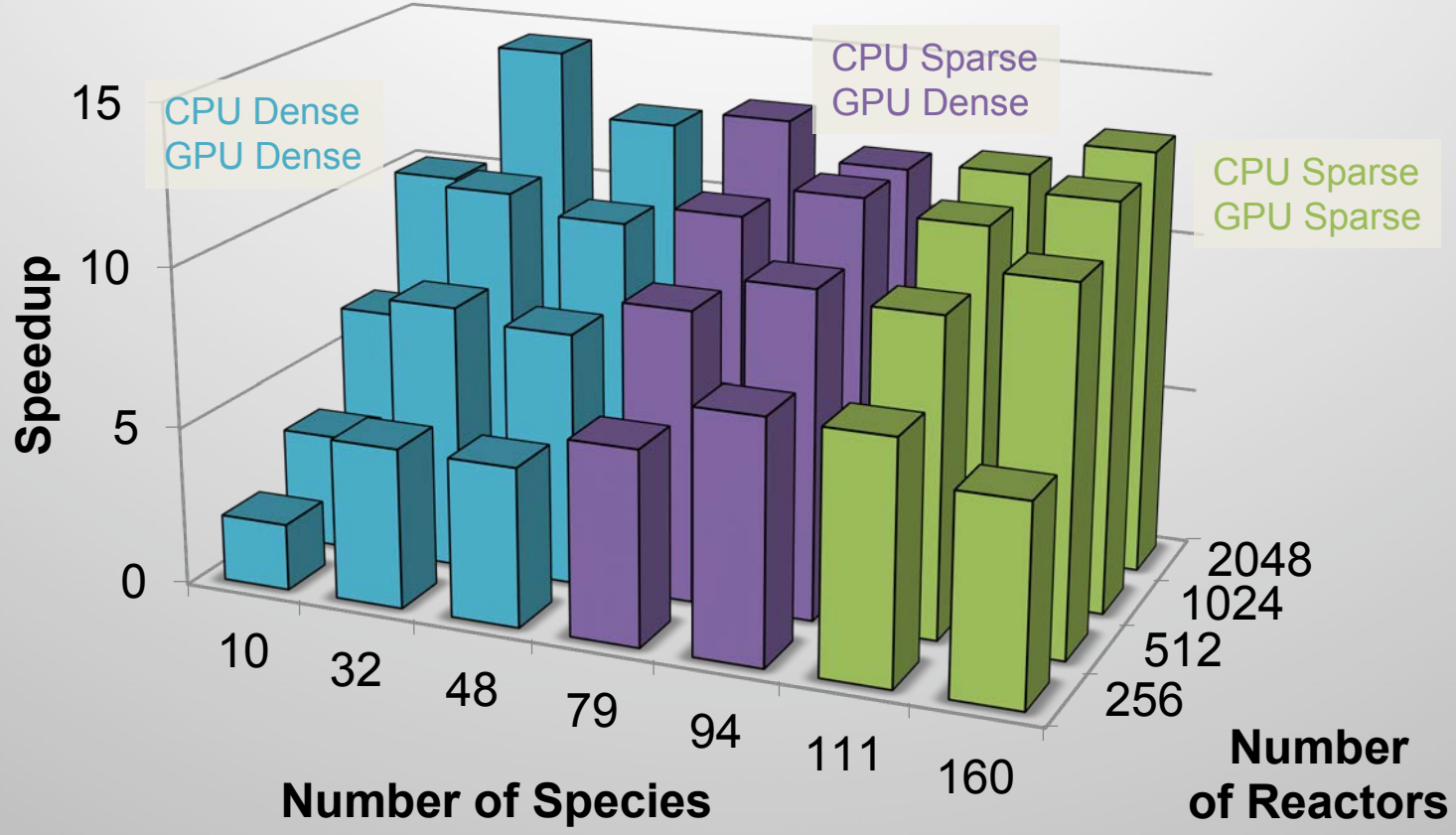
Technical Accomplishment: CPU Speedups for "Engineering" Size Mechanisms



Simulation of auto-ignition with ConvergeCFD
Comparing Converge chemistry to LLNL chemistry
Speedup for small mechanisms due to efficient calculation of chemical derivatives

~1.5-2x speedup for 10-100 species mechanisms. >2x for larger mechanisms.

Technical accomplishment: GPU speedup for "Engineering" Size Mechanisms

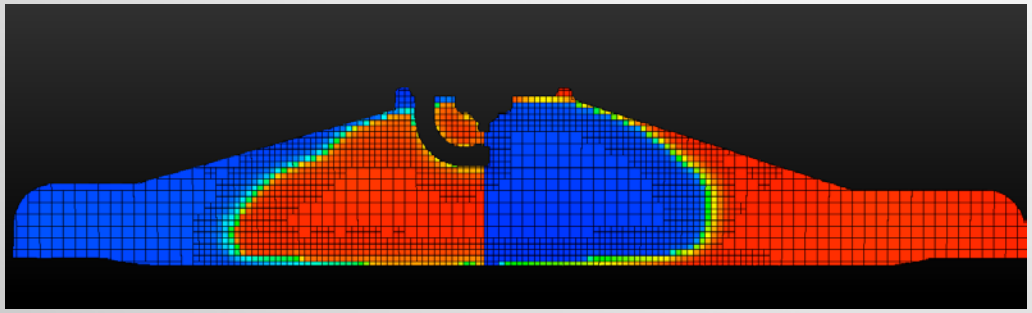


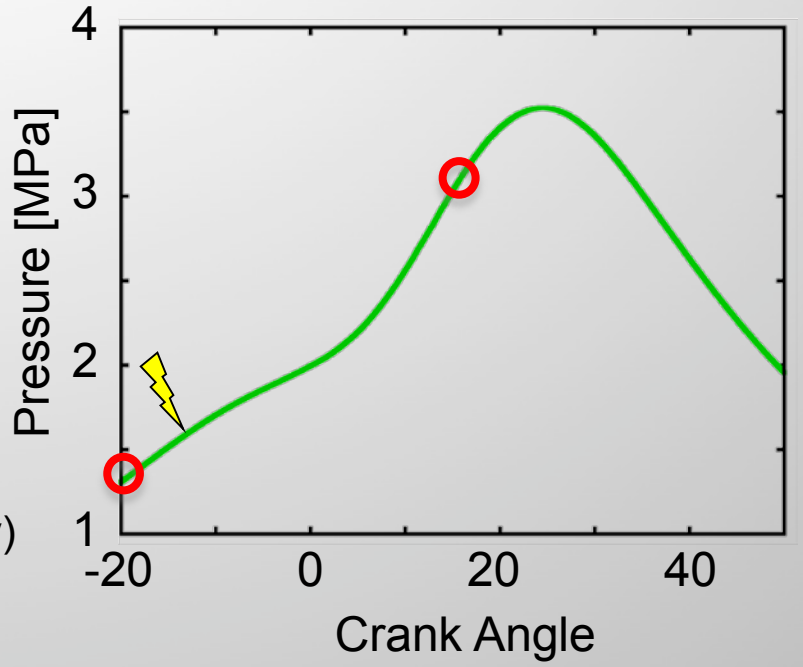
Simulation of auto-ignition
Comparing LLNL CPU chemistry to LLNL GPU chemistry
GPU Dense capability developed this FY

Results from Big Red 2 cluster at Indiana Univ.

As number of simultaneously solved reactors increases so does the speedup.

Technical Accomplishment: Engine calculation on GPU





- Compared cost of every-cell chemistry from -20 to 15 CAD (130k-350k cells with chemistry)
- 48 species iso-octane mechanism
- Highly disparate conditions in cells cut into GPU speedup
- Timing
  - 24 CPU cores = 53.8 hours vs. 24 GPU devices = 14.5 hours
  - Speedup = 53.8/14.5 = *3.7x*

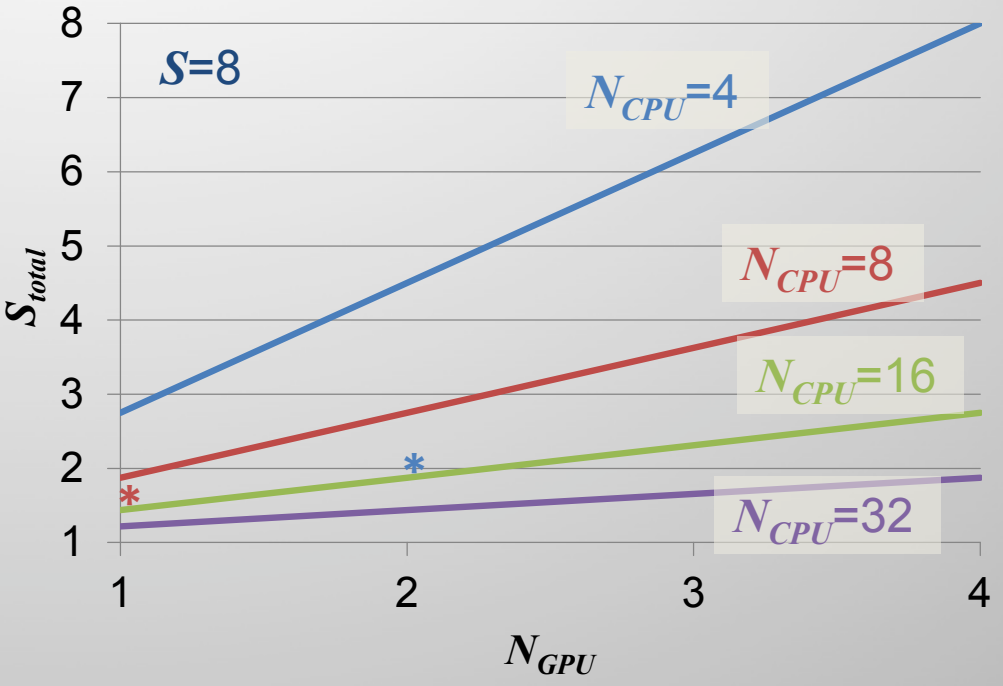
Results from Big Red 2 cluster at Indiana Univ.

Good speedup for practical engine case.

Technical Accomplishment: CPU-GPU Work-sharing

Ideal Case

$$S_{total} = \frac{\left(N_{CPU} + N_{GPU}\left(S-1\right)\right)}{N_{CPU}}$$

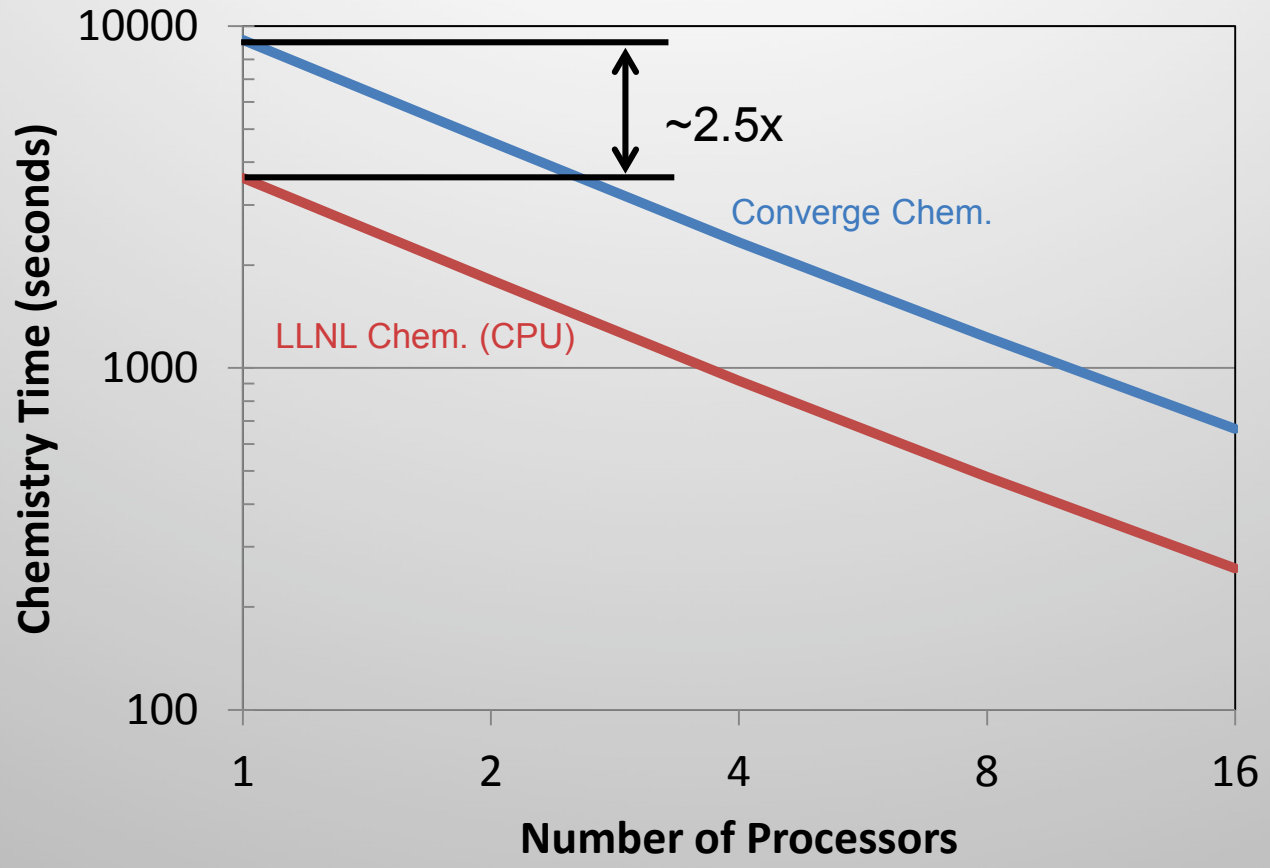


- GPU Speedup = S
- Number of CPU cores = N_{CPU}
- Number of GPU devices = N_{GPU}

*Big Red 2 (IU): 1.4375
*Surface (LLNL): 1.8750

We want to use the whole machine.

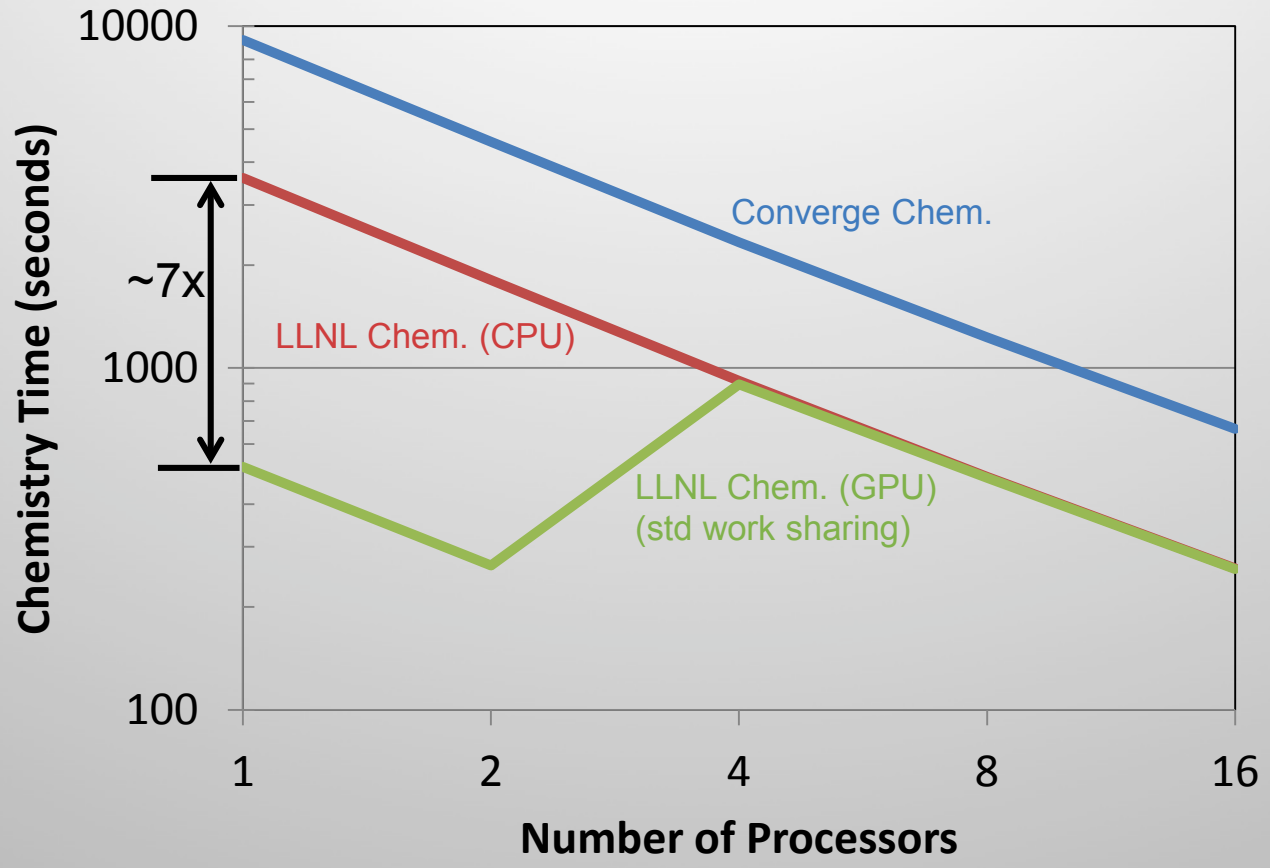
CPU-GPU Work-sharing: Strong scaling



Every cell chemistry simulation of auto-ignition; 53 species; ~10,000 cells; 16xCPU + 2xGPU

Strong scaling is good for this problem on CPU.

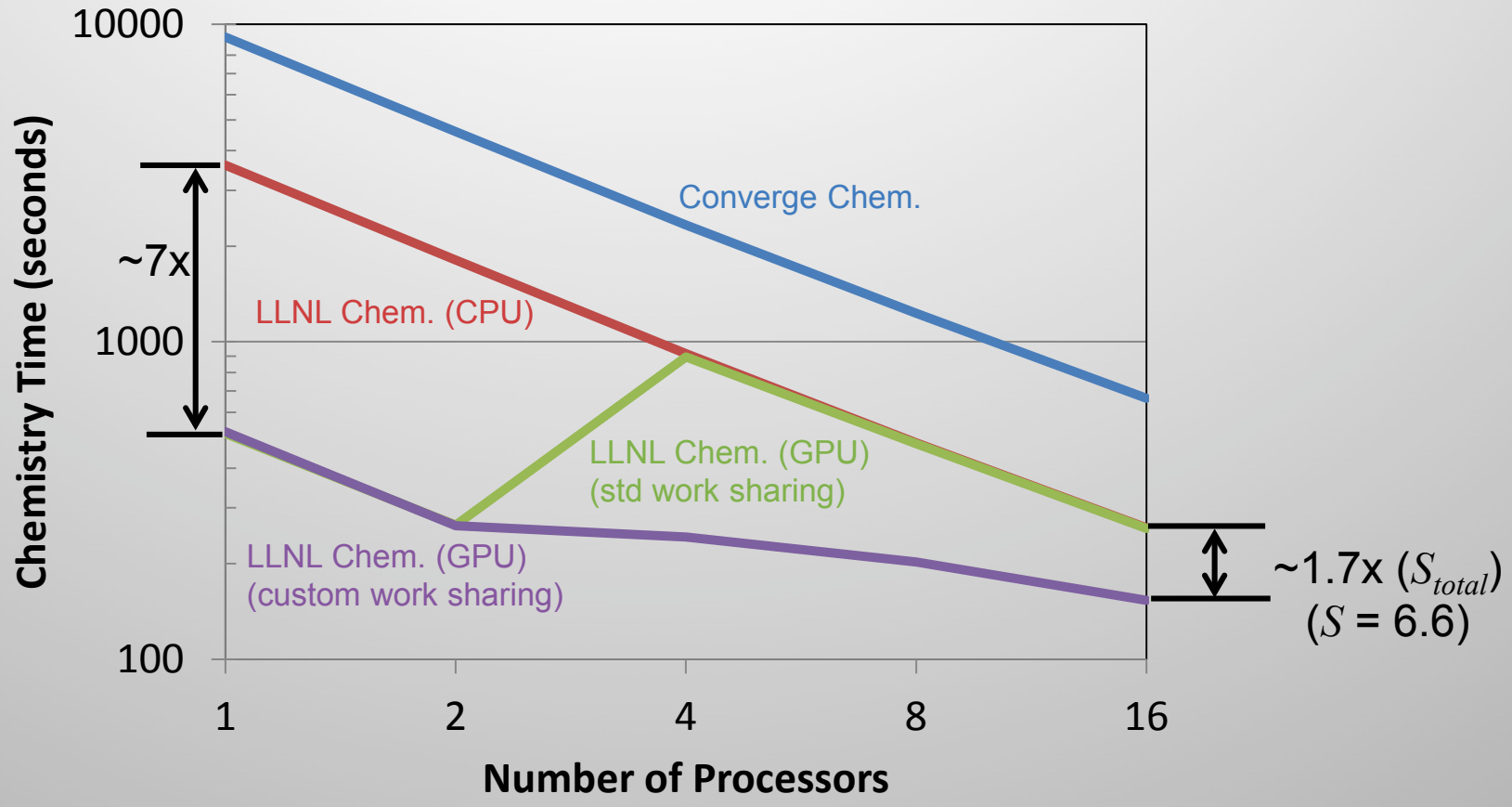
CPU-GPU Work-sharing: Strong scaling



Every cell chemistry simulation of auto-ignition; 53 species; ~10,000 cells; 16xCPU + 2xGPU

Poor scaling with GPUs, if all processors get the same amount of work.

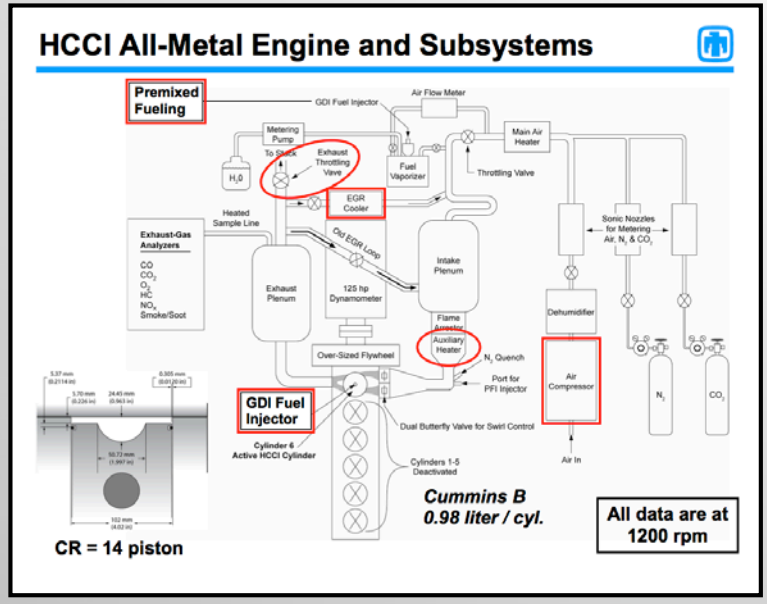
CPU-GPU Work-sharing: Strong scaling



Every cell chemistry simulation of auto-ignition; 53 species; ~10,000 cells; 16xCPU + 2xGPU

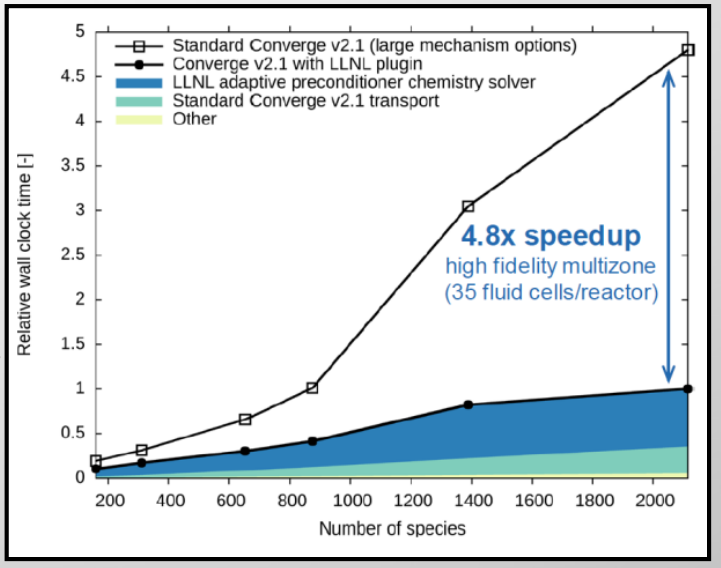
Good scaling when GPU processors given appropriate work load.

Technical Accomplishment: Uncertainty and sensitivity analysis of experimental measurements and simulation results for HCCI engine performance.



Experiments

Uncertainty Quantification



Simulations w/ Advanced Numerics + HPC

Goals of this work:

- Quantify computational model accuracy, using validation metrics from statistics
- Provide in-depth sensitivity characterization as a function of broad ranges of inputs
- Help the combustion community identify most relevant research paths
- Model Validation

Full Comparison of Well Characterized Experiments with Simulations

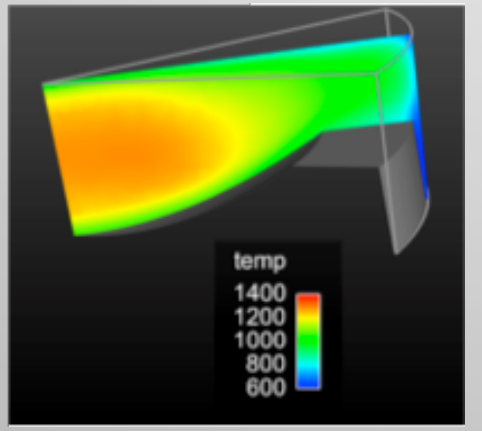
Technical Accomplishment: Uncertainty and sensitivity analysis of experimental measurements and simulation results for HCCI engine performance.

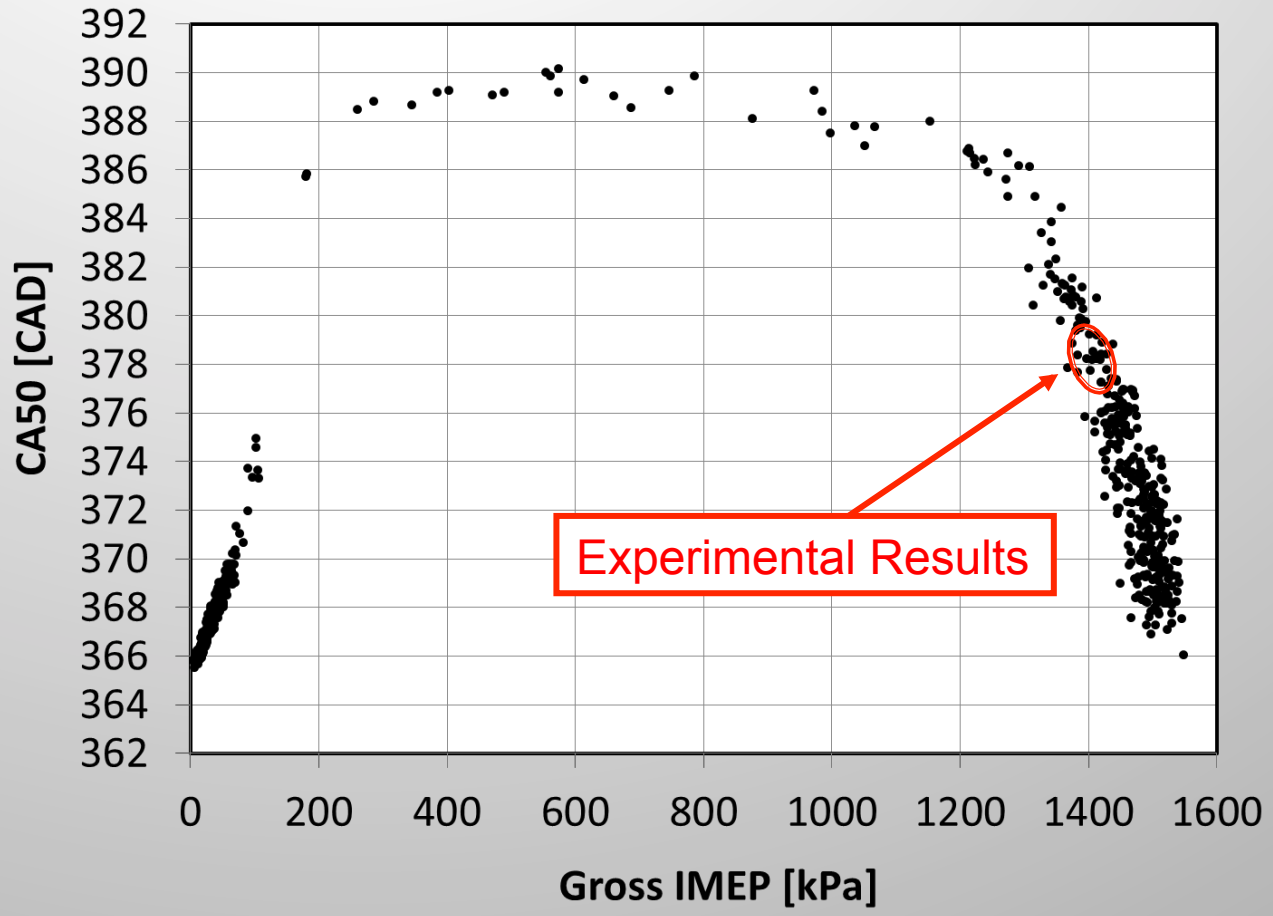
	Variable	Mean	Uncertainty	Distribution	Comments
Composition	*Mass of fuel [g]*	0.0731	±0.002	N, 95% C.I.	Uncertainty estimated from flow meter and RPM
	O_2 [%]	10.5	±0.05	N, 95% C.I.	Uncertainty estimated
	Water removal [%]	10-90	10-90	Uniform	No data available, uniform between 10 and 90%
	Residuals [%]	3	±1.5	Triangular	No data available, triangular centered on 3 %
	Combustion efficiency [%]	98.7	±1	Triangular	Estimated, triangular centered on 98.7 %
Thermo	*IVC Pressure [bar]*	2.4	±0.056	N, 95% C.I.	Calculated from transducer's specs
	IVC Temperature [K]	390-410	390-410	Uniform	Estimated from prelim runs
	Wall temperature [K]	390-460	390-460	Uniform	Estimated
	Initial tke [m^2/s^2]	41	10 to 166	Triangular	Estimated
	Swirl Profile [-]	3.11	0 to 3.86	Triangular	0 and 3.83, typical 3.11 (from Converge Manual)
	Initial swirl ratio [-]	*0.93*	0 to 1	Triangular	0 and 1, typical 0.93 (from Converge Manual)
Geometry	*Engine speed [RPM]*	1200	± 24	N, 95% C.I.	Estimated
	IVC Crank angle [CAD]	-155	±0.05	N, 95% C.I.	Tunelstal, 2009
	Stroke [m]	0.12	±2.500E-05	N, 95% C.I.	Estimated, typical engineering requirements
	Connecting rod [m]	0.192	±2.500E-05	N, 95% C.I.	Estimated, typical engineering requirements

Qualitative sub-models: surrogate composition and 679 species kinetic mechanism from Mehl *et al* (LLNL), Redlich-Wong equations of state, Angelberger wall heat transfer model, RNG k-eps turbulence model

Technical Accomplishments: Array of simulations captures the variability in computed results

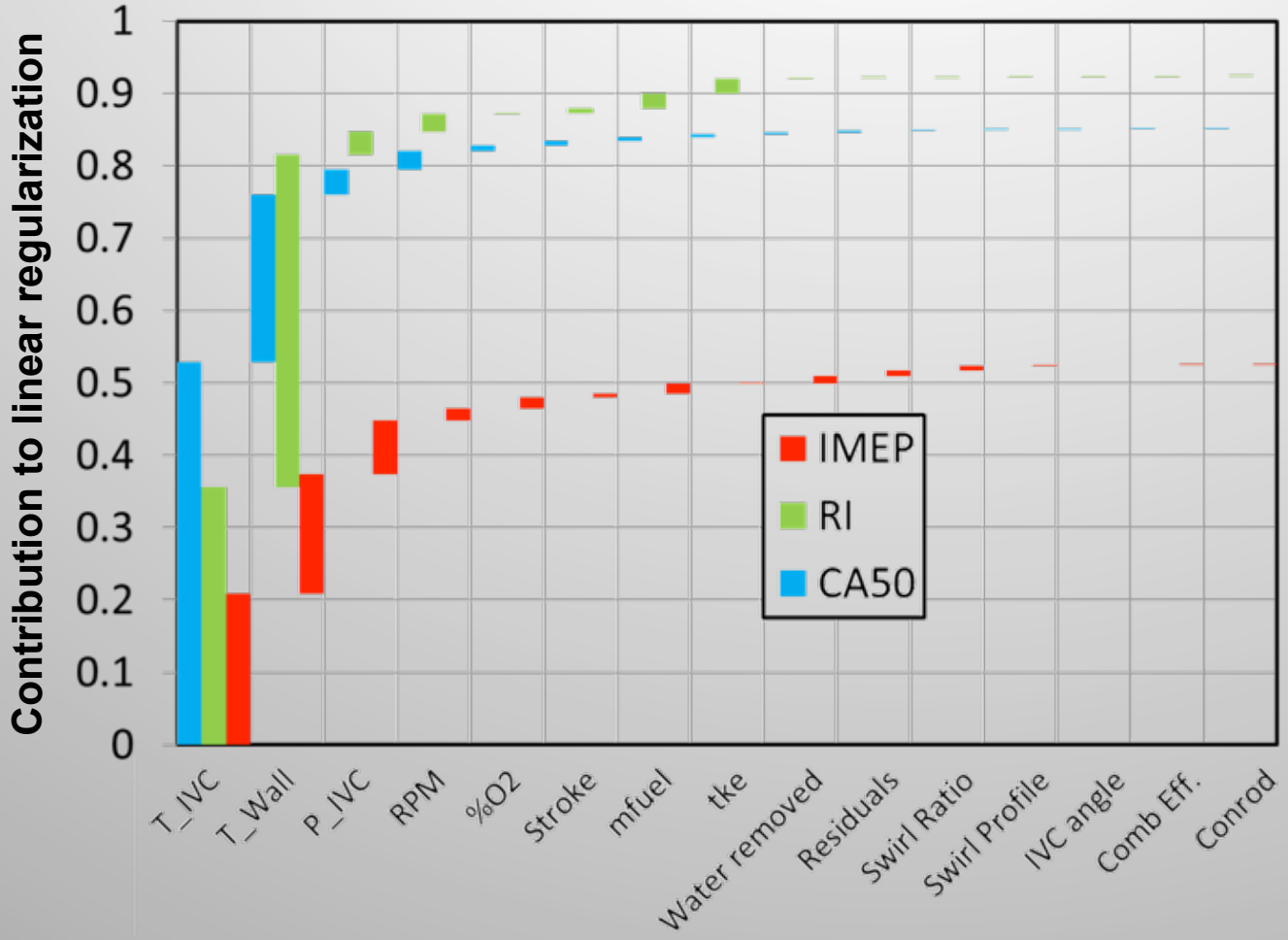
~ 3 hours per run (24 procs)
(w/ LLNL chem. solver)
~20,000 cells @ IVC
1,000 runs
~ 60% ignited





Uncertainty Propagation through computational model shows large variance of the outputs

Technical accomplishment: Statistical learning methods enables characterization of the sensitivity of each simulation output to each input



Subset selection method

For the 3 main outputs, highest sensitivity to:

T_IVC
T_Wall
P_IVC

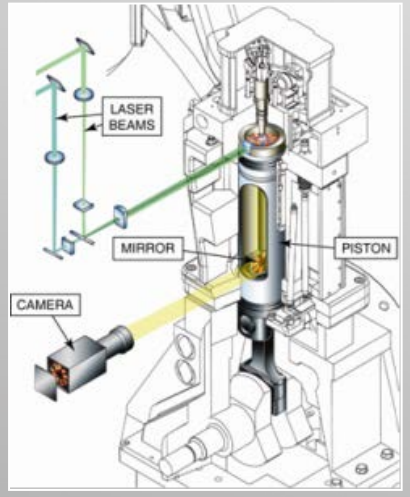
Then

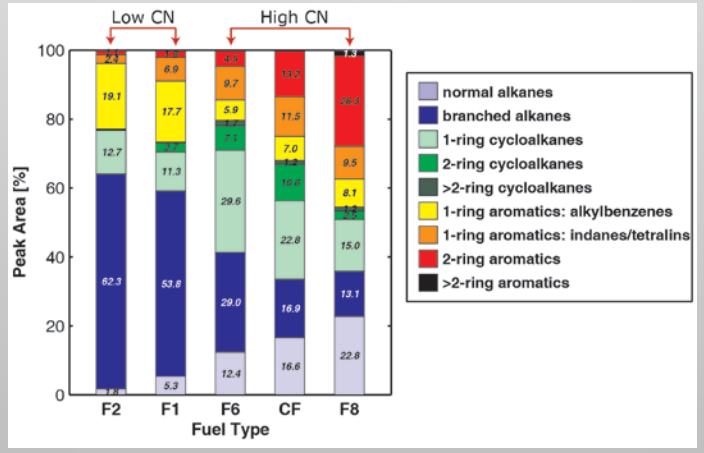
RPM
$\%O_2$
Sroke
mfuel

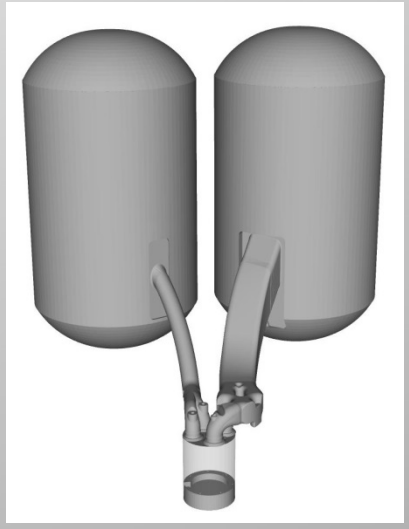
Prelim SA shows that better characterization of T and P at IVC and T_wall is necessary

Technical accomplishment/progress: Simulations of Sandia Heavy Duty Optical Diesel Engine

- Engine fueled with diesel and surrogate
- Simulations can provide detailed kinetic information of different fuel mixtures relative to engine performance.
- Leveraging accomplishments under Fuels program.







C. Mueller & co-workers heavy duty diesel engine @ SNL

3D CONVERGE model

FY2014 Reviewer's comments and our response

- Mostly positive comments in all categories.
- Licensing/Availability:
  - " … [R]eviewer noted that the PI also showed a chart in Slide 12 showing the linkage of the advanced chemistry algorithms with commercial and open-source codes. The reviewer asked if the PI could explain the following: how the license agreement works; how this interplays with linking the combustion algorithms with other codes; if this capability was being shared with ANL because they investigate high mesh resolutions for their applications; …"
  - Beta testing at multiple sites; targeting a streamlined, tiered approach to licensing
- Experimental Validation:
  - "… disappointing that there still existed little validation …"
    "… more effort should be spent addressing validation …"
    "… would like to see stronger application of the combustion approach to engine validation cases …"
  - We are taking this concern to heart with our work on sensitivity and uncertainty analysis for HCCI/PCCI. We are also working on getting the tools to collaborators for them to apply to a broad range of problems.

We appreciate the reviewer's guidance and are striving to meet their requests.

Collaboration – We have ongoing interactions with industry, national laboratories, and universities

- **Advanced Engine Combustion (AEC)** working group (Industry, National labs, Univ. of Wisc., Univ of Mich., MIT, UC Berkeley): semiannual meetings and informal collaboration
- **Cummins**: GPU CRADA under review for July 1 start. CPU/GPU solvers for Converge CFD on Indiana Univ. GPU supercomputer.
- **General Motors**: Testing CPU solver package for ConvergeCFD engine simulations
- **Convergent Science Inc. (CSI)**; Multi-zone model development, thermo-chemical functions (CPU/GPU), adaptive preconditioners (CPU/GPU).
- **NVIDIA:** Hardware, software and technical support for GPU chemistry development
- **Universities**: UC Berkeley, Univ. Wisconsin, Clemson Univ., SFSU
- **Sandia National Laboratory**: engine experiments
- **Fuels for Advanced Combustion Engines (FACE)** working group

We collaborate broadly and are eager for interaction with interested groups.

Remaining Challenges and Barriers

- ***Detailed chemistry in CFD is still expensive***
  - ***Real fuel mechanisms are large***
  - ***Prediction of kinetically controlled ignition and emissions requires fine detail***
- ***Coupling of chemical kinetics with sprays and soot formation***

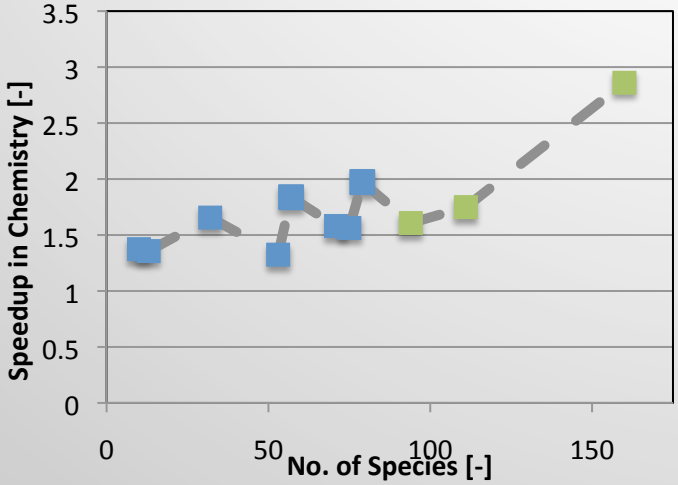
We will address these issues in our future work.

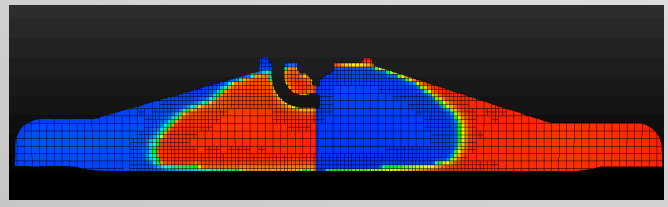
Future work: Improve physical accuracy and computational efficiency of engine combustion modeling tools

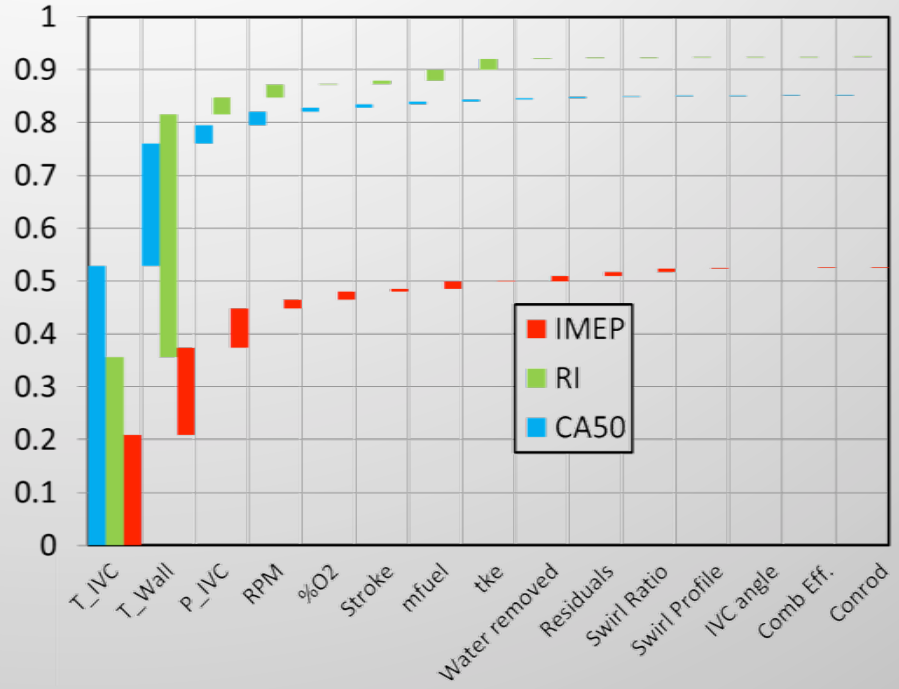
- ***Improved parallel CFD with detailed chemistry***
  - Continue to improve chemistry solver performance in engine CFD
    - General:
      - Improve parallel work balancing algorithms
      - Alternate integration methods
    - GPU:
      - Custom code generation for mechanism RHS
      - Multi-precision algorithms
- ***Engine simulation with LLNL parallel CFD with chemistry***
  - Extend uncertainty analysis to include effects of kinetics/sprays.
  - Continue simulations probing effects of fuel kinetics on diesel combustion.
- ***Continue technology transfer and licensing activities***

We continue improving model efficiency and accuracy.

Summary: We are providing industry and researchers with accurate and efficient engine combustion modeling tools





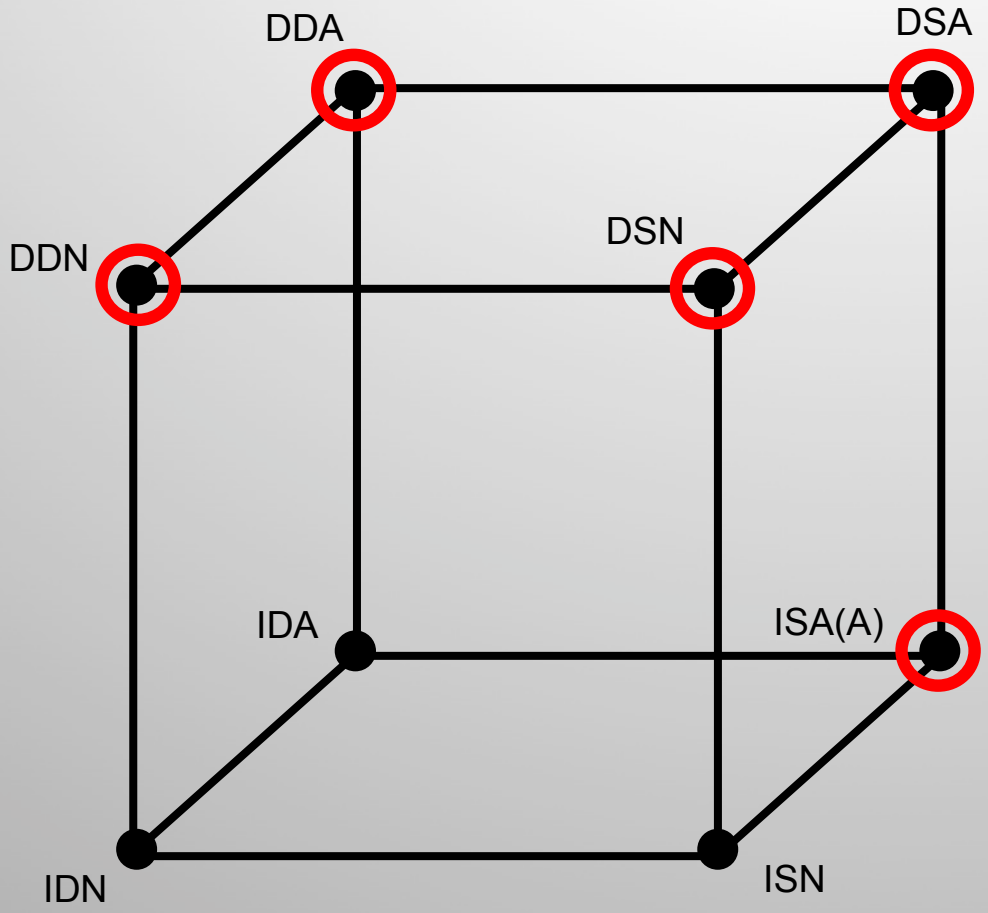


- Fast detailed chemistry for CFD:
  - Across the board speedups for CPU and GPU enabled chemistry
  - Improvements to interface including work balancing
- Broad quantification of experimental and simulation uncertainty and sensitivity for HCCI
- Continuing focus on reducing time to results for engine simulations and proving applicability of the tools to modern engine combustion concepts.

Thank You!

Technical Back-Up Slides

Three major variables of implicit ODE integration methods



Linear Solver:
Direct (D)
Iterative (I)

Matrix Storage:
Dense (D)
Sparse (S)

Jacobian Formation:
Analytic (A)
Numerical (N)

Implemented and tested for small to medium size mechanism

Different tradeoffs in terms of computational efficiency, but not accuracy.

Equations solved for integrating detailed chemistry

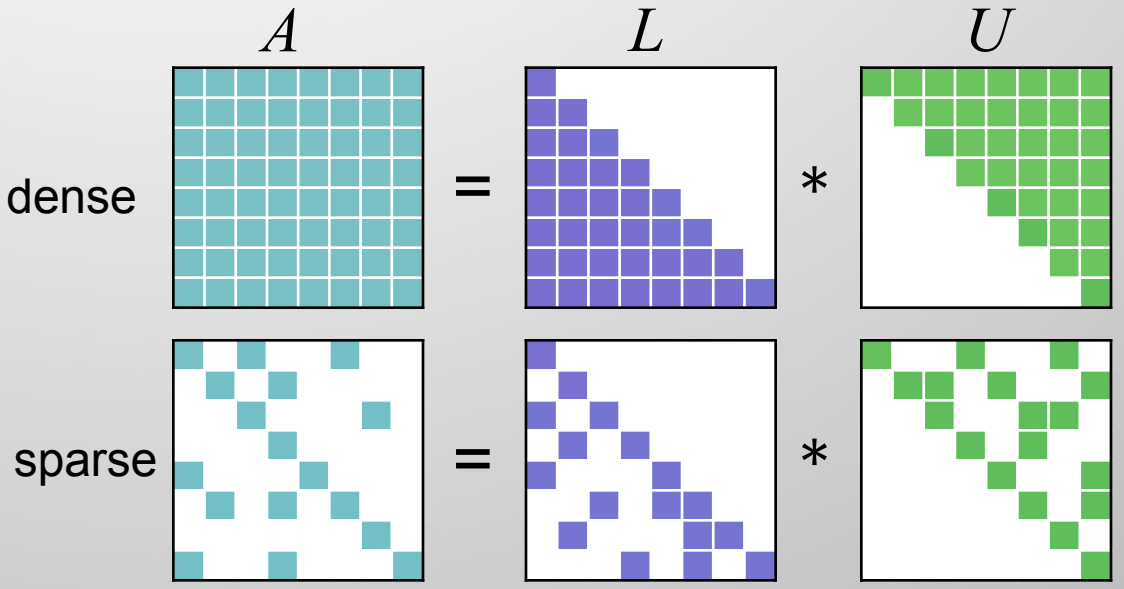
Derivative Equations
(vector calculations)

$$\frac{dy_i}{dt} = \frac{w_i}{\rho}\frac{dC_i}{dt}$$

$$\frac{dT}{dt} = -\frac{RT}{\rho c_v}\sum_{i}^{species} u_i \frac{dC_i}{dt}$$

Derivative represents system of equations to be solved (perfectly stirred reactor).

Jacobian Matrix Solution



- Matrix solution required due to stiffness
- Matrix storage in dense or sparse formats

Significant effort to transform fastest CPU algorithms to GPU appropriate versions.

Technical Hurdle: Chemistry not well suited to direct implementation on GPU

Net rates of production

$$\frac{dC_i}{dt} = \sum_{j}^{create} R_j - \sum_{j}^{destroy} R_j$$

- Chemical species connectivity
- Generally sparsely connected
- Leads to poor memory locality
- Bad for GPU performance

Chemical reaction rates of

$$R_i = k_i \prod_{j}^{species} C_j^{\nu_{ij}}$$

Chemical reaction step rate coefficients

Arrhenius Rates

$$k_i = A_i T^{n_i} e^{-\frac{E_{A,i}}{RT}}$$

Equilibrium Reverse Rates

$$k_i = k_{i,f} K_{eq} = k_{i,f} \exp\left(\sum_{j}^{prod} \frac{G_j^0}{RT} - \sum_{j}^{reac} \frac{G_j^0}{RT} \right)$$

Third-body enhanced Rates

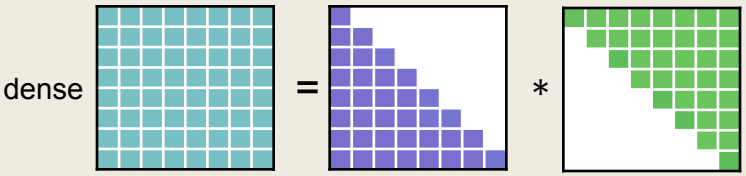
$$k_i' = k_i \sum_{j}^{species} \alpha_j C_j$$

Fall-off rates

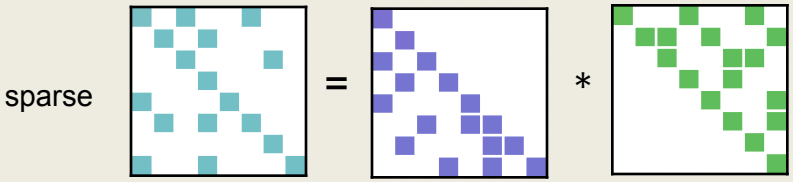
$$k_i' = k_i \ldots$$

Species Production Rates: Major component of derivative; Lots of sparse operations.

Matrix Solution Methods

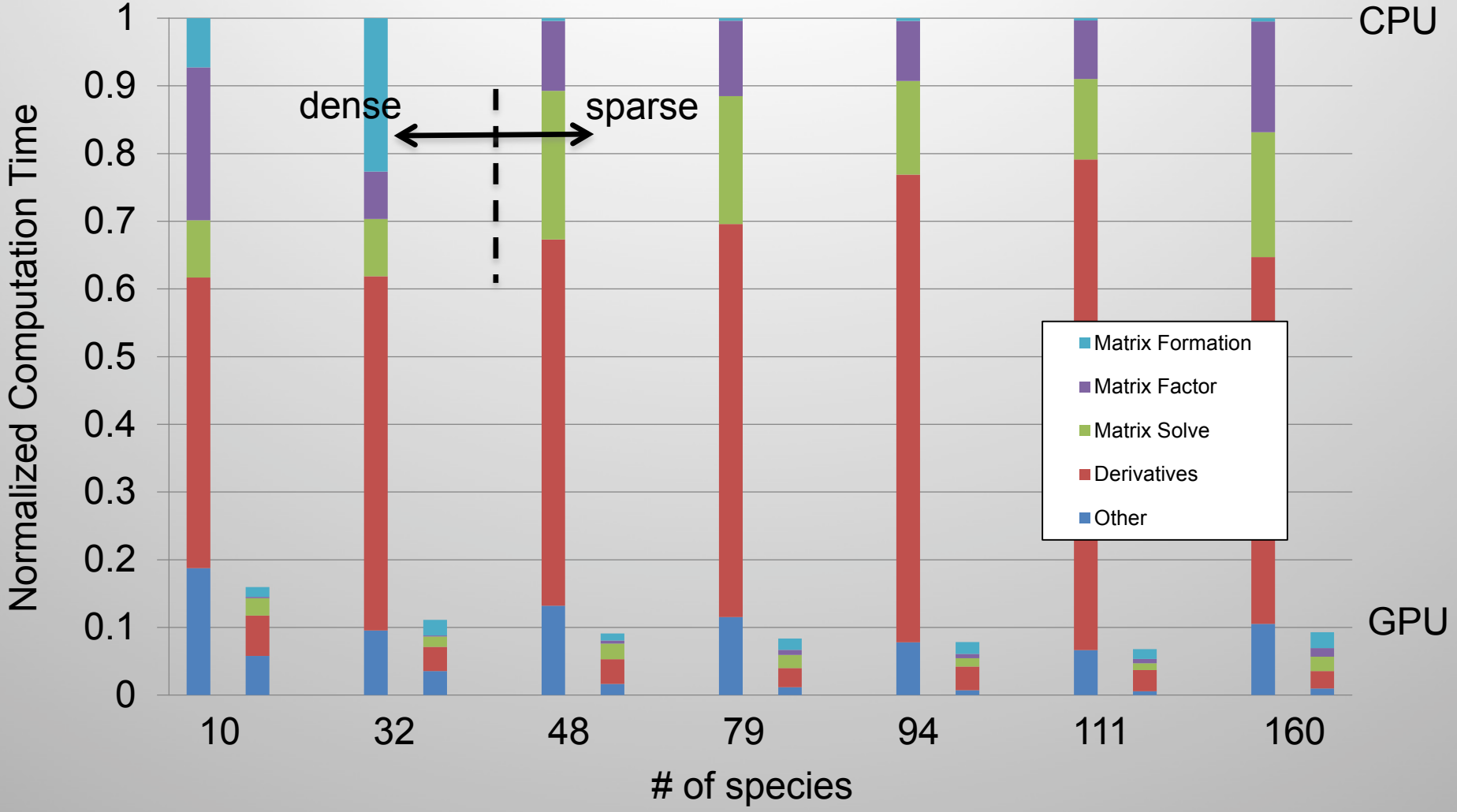


- CPU
  - LAPACK
    - dgetrf
    - dgetrs
- GPU
  - CUBLAS
    - dgetrfbatched
    - dgetribatched
    - batched matrix-vector multiplication



- CPU
  - SuperLU
    - dgetrf
    - dgetrs
- GPU
  - GLU (soon cusolverRF (7.0))
    - LU refactorization
      (SuperLU for first factor)
    - LU solve
    - Conglomerate matrix (<6.5)
    - Batched matrices (>= 6.5)
      (*2-4x faster*)

Cost Breakdown for Chemical Integration on CPU and on GPU



Costs evenly distributed across compute tasks both on CPU and GPU